

```

VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLL      IIIIIIIII  000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLL      IIIIIIIII  000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLL      IIIIIIIII  000000000000
VVV      VVV  MMMMMM  MMMMMM  SSS      LLL      III      000      000
VVV      VVV  MMMMMM  MMMMMM  SSS      LLL      III      000      000
VVV      VVV  MMMMMM  MMMMMM  SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSS  LLL      III      000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSS  LLL      III      000000000000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSS      LLL      III      000      000
VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLLLLLLLLLLLLLLLL  IIIIIIIII  000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLLLLLLLLLLLLLLLL  IIIIIIIII  000000000000
VVV      VVV  MMM      MMM      SSSSSSSSSSSSS  LLLLLLLLLLLLLLLLL  IIIIIIIII  000000000000

```

```
SSSSSSSS TTTT TTTT AAAA AAAA RRRRRRRR DDDDDDDD EEEEEEEEE FFFFFFFF MM MM PPPPPPPP
SSSSSSSS TTTT TTTT AAAA AAAA RRRRRRRR DDDDDDDD EEEEEEEEE FFFFFFFF MM MM PPPPPPPP
SS      TT      AA      AA      RR      RR      DD      DD      EE      FF      MMMM  MMMM  PP      PP
SS      TT      AA      AA      RR      RR      DD      DD      EE      FF      MM  MM  MM  PP      PP
SS      TT      AA      AA      RR      RR      DD      DD      EE      FF      MM  MM  MM  PP      PP
SSSSSS      TT      AA      AA      RRRRRRRR DD      DD      EEEEEEEE FFFFFFFF MM  MM  MM  PPPPPPPP
SSSSSS      TT      AA      AA      RRRRRRRR DD      DD      EEEEEEEE FFFFFFFF MM  MM  MM  PPPPPPPP
SS      TT      AA      AA      RR      RR      DD      DD      EE      FF      MM  MM  MM  PP
SS      TT      AA      AA      RR      RR      DD      DD      EE      FF      MM  MM  MM  PP
SSSSSSSS      TT      AA      AA      RR      RR      DDDDDDDD DDDDDDDD EEEEEEEEE FFFFFFFF MM  MM  MM  PP
SSSSSSSS      TT      AA      AA      RR      RR      DDDDDDDD DDDDDDDD EEEEEEEEE FFFFFFFF MM  MM  MM  PP

SSSSSSSS DDDDDDDD LL
SSSSSSSS DDDDDDDD LL
SS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SSSSSS      DD      DD      LL
SSSSSS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SS      DD      DD      LL
SSSSSSSS DDDDDDDD LLLLLLLLLL
SSSSSSSS DDDDDDDD LLLLLLLLLL
```

Version: 'V04-000'

(\* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
(\* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
(\* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
(\* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
(\* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
(\* TRANSFERRED.

(\* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
(\* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
(\* CORPORATION.

(\*) DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

{++}

C FACILITY: VAX/VMS System Macro Libraries

**ABSTRACT:**

This file contains the SDL source for all user visible operating system interfaces from M to P.

ENVIRONMENT:

n/a

AUTHOR: The VMS Group                      CREATION DATE: 1-Aug-1976

MODIFIED BY:

V03-055 DAS0002 David Solomon 9-Jul-1984  
Add MNT\$V\_NOREBUILD to mount flags longword.

V03-054 ACG0432 Andrew C. Goldstein, 6-Jul-1984 16:34  
Add PRC\$V\_NOPASSWORD bit to \$CREPRC flags

V03-053 MHB0156 Mark Bramhall 2-May-1984  
Add SYSUAF\_COP record type/id to \$NSARECDEF.



V03-052 RLRMVER Robert L. Rappaport 26-Apr-1984  
Add MSG\$ RC25MVER, MSG\$ RDRXMVER, MSG\$ TU81MVER, and  
MSG\$ MAYAMVER symbols added to \$MSGDEF.

V03-051 TMH0051 Tim Halvorsen 12-Apr-1984  
Remove \$M symbols from PRVDEF (V03-049) for bits in  
the second longword, since SDL cannot generate masks  
for bit offsets greater than 32.

V03-050 MHB0139 Mark Bramhall 12-Apr-1984  
Add CLISPEC flag to \$PRCDEF.

V03-049 MCN0165 Maria del C. Nasr 09-Apr-1984  
Add mask values to \$PRVDEF.

V03-048 RSH0128 R. Scott Hanna 28-Mar-1984  
\$NSARECDEF Add the packet type NSASK\_PKTTP\_STATUS and  
remove the record type ACL.

V03-047 RSH0109 R. Scott Hanna 28-Feb-1984  
\$NSARECDEF Change time field in the security auditing  
record header from a longword to a quadword.

V03-046 HH0004 Hai Huang 28-Feb-1984  
Add MNT\$V\_CLUSTER for cluster-wide mount support.

V03-045 ROW0318 Ralph D. Weber 27-FEB-1984  
Add OPCOM message codes for shadow set mount verification  
messages; MSG\$\_SHAMEMFAL, member failed out of shadow set, and  
MSG\$\_SHARDUCED, shadow set reduced.

V03-044 MMD0241 Meg Dumont, 24-Feb-1984 11:15  
Add MTADEF codes to support the mag tape accessibility routine.

V03-043 RSH0097 R. Scott Hanna 02-Feb-1984  
Replace \$NSARECDEF.

V03-042 KPL0002 Peter Lieberwirth 2-Feb-1984  
Add \$PR8NNDEF for Nautilus.

V03-041 ACG0386 Andrew C. Goldstein, 10-Jan-1984 16:29  
Add PRC\$V\_PASSWORD to \$CREPRC flags

V03-040 RLRPR8SS2 Robert L. Rappaport 9-Dec-1983  
Additional minor corrections to \$PR8SSDEF.

V03-039 DAS0001 David Solomon 29-Nov-1983  
Add MNT\$\_JRNLRECORD\_SIZE for specifying max journal recordsize  
on MOUNT.

V03-038 RLRPR8SS1 Robert L. Rappaport 28-Nov-1983  
Modify \$PR8SSDEF according to new Scorpio spec.

V03-037 RLRPR8SS Robert L. Rappaport 11-Nov-1983  
Add \$PR8SSDEF for Scorpio specific registers.

V03-036 KPL0001 Peter Lieberwirth 8-Nov-1983  
Add PR\$\_SID\_TYP8SS for Scorpio, PR\$\_SID\_TYP8NN for Nautilus.

V03-035 TMK0001 Todd M. Katz 27-Oct-1983  
Add the process quota list code PQL\$\_JTQUOTA.

V03-034 KDM0075 Kathleen D. Morse 23-Aug-1983  
Update PR\$\_TYPMAX to 8.

V03-033 CWH1011 CW Hobbs 18-Aug-1983  
Add MSG\$\_CLUMBX and MSG\$\_TM78MVER messages.

V03-032 KDM0067 Kathleen D. Morse 4-Aug-1983  
Add processor-specific IPR macros for Micro-VAX, \$PRUV1DEF  
and \$PRUV2DEF.

V03-031 SBL0031 Steve Lionel 29-Jul-1983  
Add comment to \$PRVDEF about updating [RTL.SRC]LIBLEXICA.B32.

V03-030 WMC0030 Wayne Cardoza 28-Jul-1983  
PRCDEF item codes for logical name attributes.

V03-029 MMD0191 Meg Dumont, 28-Jul-1983 9:49  
Changed bit in VCBDEF from AUTO to NOAUTO to make mag tape  
AVL/AVR consistent between DCL and MOUNT system service

V03-028 KDM0050 Kathleen D. Morse 15-Jul-1983  
Add cpu-dependent IPR definitions: \$PR730DEF, \$PR750DEF,  
\$PR780DEF, and \$PR790DEF. Remove ACCS, ACCR, PME, TODR,  
ICR, and NICR from \$PRDEF and add to cpu-dependent IPRs.

V03-027 MLJ0114 Martin L. Jack 22-Jun-1983  
Add MSG\$\_GETQUI.

V03-026 RSH0035 R. Scott Hanna 16-Jun-1983  
Provide permanent fix for \$NSARECDEF symbols.

V03-025 ADE9001 A. Eldridge 27-May-1983  
Temporary modifications to \$NSARECDEF to allow build  
to proceed.

V03-025 RSH0023 R. Scott Hanna 24-May-1983  
Add \$NSARECDEF (Security Auditing record definitions)

V03-024 RSH0018 R. Scott Hanna 21-May-1983  
Add SECURITY privilege to \$PRVDEF

V03-023 KDM0046 Kathleen D. Morse 20-May-1983  
Add Micro-VAX cpu definitions to \$PRDEF.

V03-022 PRB0172 Paul Beck 26-Apr-1983  
Add TMPJNL and PRMJNL privileges

V03-021 WMC0017 Wayne Cardoza 10-Apr-1983  
Add IMGDMF flag to PRCDEF



V03-020 MMD0111 Meg Dumont, 25-Mar-1983 9:51  
Added MNTDEF bits for new mount qualifiers

V03-019 WMC0016 Wayne Cardoza 08-Mar-1983  
Add item codes to PRCDEF

V03-018 WMC0015 Wayne Cardoza 06-Mar-1983  
Add PRC\$V\_INTER and PRC\$V\_DETACH

V03-017 LMP0083 L. Mark Pilant, 28-Feb-1983 9:54  
Add a blurb that indicates what modules must be changed when  
privileges are added.

V03-16 LMP0082 L. Mark Pilant, 28-Feb-1983 8:41  
Add difinitions for the following privileges: UPGRADE,  
DOWNGRADE, GRPPRV, and READALL.

V03-015 JLV0233 Jake VanNoy 24-FEB-1983  
Add definition for SHARE privilege.

V03-014 WMC0014 Wayne Cardoza 04-Jan-1983  
Add 790 scratchpad registers to PRDEF.

V03-013 RLRDENS Robert L. Rappaport 21-Dec-1982  
Add density support bit fields to MTDEF. These bits  
will tell what densities are supported on a drive.

V03-012 TCM0004 Trudy C. Matthews 13-Dec-1982  
Add PR\$\_STXCS and PR\$\_STXDB definitions.

V03-011 ACG0303 Andrew C. Goldstein, 9-Dec-1982 16:06  
Add FILL attribute to extraneous field names

V03-010 RLRSPEEDA Robert L. Rappaport 09-Nov-1982  
Corrected Speed definition in MTDEF.

V03-009 TCM0003 Trudy C. Matthews 02-Nov-1982  
Changed PR\$\_SID\_TYP7VV to PR\$\_SID\_TYP790.

V03-008 RLRSPEED Robert L. Rappaport 21-Oct-1982  
Add Speed field and speed values to MTDEF.

V03-007 MLJ0097 Martin L. Jack, 9-Sep-1982 16:38  
Add MSG\$\_SNDJBC.

V03-006 RLRSEREX Robert L. Rappaport 26-Aug-1982  
Remove MTSM\_CLSEREXCP.

V03-005 STJ0320 Steven T. Jeffreys 25-Aug-1982  
Add support for recovery unit journalling in \$MOUNT.

V03-004 RLR0001 Robert L. Rappaport 4-Aug-1982  
Add serious exception bits to MTDEF

V03-003 TCM0002 Trudy C. Matthews 28-Jul-1982  
Replace 11/790-specific Internal Processor Register definitions.

V03-002 STJ0307 Steven T. Jeffreys 18-May-1982  
Added MNT\$V\_NOUNLOAD.  
V03-001 RLRV3A1 Robert L. Rappaport 5-Apr-1982  
Added MSG\$\_UDA50MVER and MSG\$\_DUPUNITNO.

STA  
mod  
/++  
/++  
/++  
con  
agg  
end  
end

module \$MNTDEF;

/\*+

/\*

/\* FLAG BITS FOR THE \$MOUNT SYSTEM SERVICE.

/\*

/\*-

aggregate MNTDEF union prefix MNT\$;

MNTDEF BITS structure fill;

FOREIGN bitfield mask;

GROUP bitfield mask;

NOASSIST bitfield mask;

NODISKQ bitfield mask;

NOHDR3 bitfield mask;

NOLABEL bitfield mask;

NOWRITE bitfield mask;

OVR\_ACCESS bitfield mask;

OVR\_EXP bitfield mask;

OVR\_IDENT bitfield mask;

OVR\_SETID bitfield mask;

READCHECK bitfield mask;

SHARE bitfield mask;

MESSAGE bitfield mask;

SYSTEM bitfield mask;

WRITECHECK bitfield mask;

WRITETHRU bitfield mask;

NOCACHE bitfield mask;

OVR\_LOCK bitfield mask;

NOMNTVER bitfield mask;

NOUNLOAD bitfield mask;

NOJRNL bitfield mask;

NEWJRNL bitfield mask;

NOAUTO bitfield mask;

INIT\_ALL bitfield mask;

INIT\_CONT bitfield mask;

OVR\_VOLO bitfield mask;

INTERCHG bitfield mask;

CLUSTER bitfield mask;

NOREBUILD bitfield mask;

end MNTDEF\_BITS;

/\*

/\* Item codes for mount parameters.

/\*

constant(

DEVNAM

, VOLNAM

, LOGNAM

, FLAGS

, ACCESSED

, PROCESSOR

, VOLSET

, BLOCKSIZE

, DENSITY

/\* FOREIGN OPTION SELECTED

/\* GROUP OPTION SELECTED

/\* NOASSIST OPTION SELECTED

/\* NODISKQ OPTION SELECTED

/\* NOHDR3 OPTION SELECTED

/\* NOLABEL OPTION SELECTED

/\* NOWRITE OPTION SELECTED

/\* OVERRIDE ACCESSIBILITY OPTION SELECTED

/\* OVERRIDE EXPIRATION OPTION SELECTED

/\* OVERRIDE VOLUME LABEL

/\* OVERRIDE VOLUME SET IDENT OPTION SELECTED

/\* READCHECK OPTION SELECTED

/\* SHARE OPTION SELECTED

/\* ALLOW \$MOUNT TO PRINT MESSAGES

/\* SYSTEM OPTION SELECTED

/\* WRITECHECK OPTION SELECTED

/\* WRITETHRU OPTION SELECTED

/\* TURN OFF ALL CACHING

/\* OVERRIDE AUTOMATIC WRITE-LOCK

/\* DISABLE MOUNT VERIFICATION

/\* DO NOT UNLOAD VOLUME AT DISMOUNT

/\* DO NOT ACTIVATE RECOVERY UNIT JOURNAL FILE

/\* CREATE A NEW RECOVERY UNIT JOURNAL FILE

/\* DO NOT SET THE MTAACP INTO AVR AND AVL MODE

/\* INITIALIZE ALL VOLUMES IN SET BEFORE WRITING

/\* INITIALIZE CONTINUATION VOLUMES BEFORE WRITING

/\* OVERRIDE VOL1 VOLUME IDENTIFIER FIELD

/\* VOL FOR INTERCHG NO VMS SPECIFIC INFO WRITTEN TO TAPE

/\* CLUSTER-WIDE MOUNT OPTION SELECTED

/\* DO NOT REBUILD VOLUME

/\* DEFINE CODES AS CONSTANTS

/\* DEVICE NAME

/\* VOLUME NAME

/\* LOGICAL NAME

/\* MOUNT FLAGS

/\* ACCESSED VALUE

/\* PROCESSOR NAME

/\* VOLUME SET NAME

/\* BLOCKSIZE VALUE

/\* TAPE DENSITY VALUE



```
      , EXTENT                      /* NUMBER OF EXTENT CACHE ENTRIES
      , FILEID                      /* FILE ID CACHE SIZE
      , LIMIT                       /* EXTENT CACHE LIMIT
      , OWNER                       /* VOLUME OWNER UIC
      , VPROT                      /* VOLUME PROTECTION
      , QUOTA                      /* QUOTA CACHE SIZE
      , RECORDSIZ                  /* RECORD SIZE VALUE
      , WINDOW                     /* NUMBER OF WINDOWS
      , EXTENSION                  /* DEFAULT FILE EXTENSION
      , VISUAL_ID                  /* VISUAL IDENTIFICATION
      , COMMENT                    /* USER COMMENT
      , JRNLSIZE                   /* INITIAL JOURNAL SIZE
      , JRNLEXTEND                /* JOURNAL EXTENSION QUANTITY
      , JRNLQUOTA                 /* JOURNAL BYTE QUOTA (PER R.U.)
      , JRNLRECORD_SIZE           /* JOURNAL MAXIMUM RECORD SIZE
    } equals 1 increment 1 prefix MNT tag $;
end MNTDEF;
end_module $MNTDEF;
```

```
module $MSGDEF;
```

```
/*+
/*
/* SYSTEM WIDE MAILBOX MESSAGE TYPES
/*
/*-
```

```
constant(
    TRMUNSOLIC
    , CRUNSOLIC
    , DELPROC
    , SNDSMB
    , DEVOFFLIN
    , TRMHANGUP
    , DEVONLIN
    , OPRQST
    , OPREPLY
    ) equals 1 increment 1 prefix MSG tag $;
```

```
constant(
    SMBINI
    , SMBDON
    , SNDACC
    , PURPROC
    , DELIMAG
    , PURIMAG
    , SYSFUNC
    , SNDJBC
    , GETQUI
    ) equals 8 increment 1 prefix MSG tag $;
```

```
constant(
    XM_DATAVL
    , XM_SHUTDN
    , XM_ATTN
    ) equals 11 increment 1 prefix MSG tag $;
```

```
constant(
    INIOPR
    , ABOOPR
    , SUSOPR
    , RESOPR
    , DELSMB
    , REQUE
    ) equals 16 increment 1 prefix MSG tag $;
```

```
constant(
    SMBRSP
    , ACCRSP
    ) equals 32 increment 1 prefix MSG tag $;
```

```
constant(
    SCANBAD
    , SCANRSP
    ) equals 40 increment 1 prefix MSG tag $;
```

```
/* DEFINE CODES AS CONSTANTS
```

```
/* UNSOLICITED TERMINAL DATA
/* UNSOLICITED CARD READER DATA
/* DELETE PROCESS
/* SEND TO SYMBIONT MANAGER
/* DEVICE OFFLINE
/* TERMINAL HANG UP
/* DEVICE ONLINE
/* OPERATOR REQUEST *** OVERLAPPED CODE ***
/* OPERATOR REPLY *** OVERLAPPED CODE ***
```

```
/* DEFINE SYMBIONT RESPONSE MESSAGES
```

```
/* SYMBIONT HAS INITED
/* SYMBIONT FINISHED
/* SEND MESSAGE TO ACCOUNTING MANAGER
/* PURGE PROCESS *** OVERLAPPED CODE ***
/* DELETE IMAGE *** OVERLAPPED CODE ***
/* PURGE IMAGE *** OVERLAPPED CODE ***
/* SYSTEM FUNCTION *** OVERLAPPED CODE ***
/* Send message to job controller
/* Get queue information (from job controller)
```

```
/* DEFINE DMC MESSAGES
```

```
/* DMC UNSOLICITED DATA
/* DMC LINE DOWN
/* DMC ATTENTION MESSAGE
```

```
/* SYMBIONT COMMAND MESSAGES
```

```
/* INITIATE PRINTING A FILE
/* ABORT PRINTING A FILE
/* PAUSE PRINTING THE FILE
/* RESUME PRINTING THE FILE
/* SYMBIONT SHOULD DELETE ITSELF
/* REQUEUE A FILE FOR PRINTING
```

```
/*
```

```
/* SYMBIONT MANAGER RESPONSE
/* ACCOUNTING MANAGER RESPONSE
```

```
/* FILE ACP MESSAGES
```

```
/* SCAN FILE FOR BAD BLOCKS
/* RESPONSE FROM FILE SCANNER
```

```
/* NETWORK ATTENTION CODES
```

```

constant(
  ABORT
  . CONFIRM
  . CONNECT
  . DISCON
  . EXIT
  . INTMSG
  . PATHLOST
  . PROTOCOL
  . REJECT
  . THIRDPARTY
  . TIMEOUT
  . NETSHUT
  . NODEACC
  . NODEINACC
  . EVTAVL
  . EVTRCVCHG
  . INCDAT
  . RESET
  . LINUP
  . LINDWN
  . EVTXTMCHG
) equals 48 increment 1 prefix MSG tag $;

```

```

constant(
  DEVOFFLINX
  . WRONGVOL
  . DEVURTLC
  . TRMBRDCST
  . MVCOMPLETE
  . MVABORTED
  . DISMOUNTED
  . UDA50MVER
  . DUPUNITNO
  . CLUMBX
  . TM78MVER
  . SHAMEMFAL
  . SHARDUCED
  . RC25MVER
  . RDRXMVER
  . TU81MVER
  . MAYAMVER
) equals 80 increment 1 prefix MSG tag $;

```

```

end_module $MSGDEF;

```

```

/* PARTNER ABORTED LINK
/* CONNECT CONFIRM
/* INBOUND CONNECT INITIATE
/* PARTNER DISCONNECTED - HANGUP
/* PARTNER EXITED PREMATURELY
/* INTERRUPT MESSAGE - UNSOLICITED DATA
/* NFW - PATH LOST TO PARTNER
/* PROTOCOL ERROR
/* CONNECT REJECT
/* THIRD PARTY DISCONNECT
/* CONNECT TIMEOUT
/* Network shutting down
/* Node has become accessible
/* Node has become inaccessible
/* Events are available to EVL
/* Event receiver database change
/* X25 INCOMING DATA
/* X25 CIRCUIT RESET
/* X25 PVC LINE UP
/* X25 PVC LINE DOWN
/* Event transmitter database change

/* MOUNT VERIFICATION MESSAGES

/* DEVICE OFFLINE
/* WRONG VOLUME IN DEVICE
/* DEVICE HAS BEEN WRITE LOCKED
/* TERMINAL BROADCAST
/* MOUNT VERIFICATION COMPLETED
/* MOUNT VERIFICATION ABORTED
/* VOLUME DISMOUNTED
/* UDA50 MICORCODE NOT UPTO REV
/* MSCP CONTROLLER - DUPLICATE UNIT !
/* CNXMGR to OPCOM messages
/* TM78 Microcode not up to rev level
/* Member failed out of shadow set
/* Shadow set reduced
/* RC25 MICORCODE NOT UPTO REV
/* RDRX MICORCODE NOT UPTO REV
/* TU81 MICORCODE NOT UPTO REV
/* MAYA MICORCODE NOT UPTO REV

```



```
module $MTADEF;
```

```
/*
```

```
/* MAGTAPE ACCESSIBILITY ROUTINE CODES
```

```
/*-
```

```
aggregate MTADEF union prefix MTAS;
```

```
/* DEFINITIONS FOR ACCESS_SPEC
```

```
constant NOCHAR equals 0 prefix MTA tag $K; /* ACCESS CHAR IS NOVALID
```

```
constant CHARVALID equals 1 prefix MTA tag $K; /* ACCESS CHAR IS VALID
```

```
/* DEFINITIONS FOR TYPE
```

```
constant INVOL1 equals 0 prefix MTA tag $K; /* INPUT A VOL1 ACCESS CODE
```

```
constant INHDR1 equals 1 prefix MTA tag $K; /* INPUT A HDR1 ACCESS CODE
```

```
constant OUTVOL1 equals 2 prefix MTA tag $K; /* OUTPUT A VOL1 ACCESS CODE
```

```
constant OUTHDR1 equals 3 prefix MTA tag $K; /* OUTPUT A HDR1 ACCESS CODE
```

```
end MTADEF;
```

```
end_module $MTADEF;
```

```
module SMTDEF;
```

```
/*+
```

```
/* MAGTAPE STATUS BITS
```

```
/*-
```

```
aggregate MTDEF union prefix MT$;
```

```
  MTDEF BITS structure fill;
```

```
    SEREXCP bitfield mask;
```

```
    FILL 1 bitfield fill prefix MTDEF tag $$;
```

```
    ENSEREXCP bitfield mask;
```

```
    PARITY bitfield mask;
```

```
    FORMAT bitfield mask length 4;
```

```
    DENSITY bitfield mask length 5;
```

```
    FILL 2 bitfield fill prefix MTDEF tag $$;
```

```
    LOGSOFT bitfield mask;
```

```
    LOGSOFTOG bitfield mask;
```

```
    BOT bitfield mask;
```

```
    EOF bitfield mask;
```

```
    EOT bitfield mask;
```

```
    HWL bitfield mask;
```

```
    LOST bitfield mask;
```

```
    SUP_NRZI bitfield mask;
```

```
    SUP_PE bitfield mask;
```

```
    SUP_GCR bitfield mask;
```

```
    SPEED bitfield mask length 8;
```

```
  end MTDEF_BITS;
```

```
/* SERIOUS EXCEPTION PRESENT
```

```
/* SPARE UNUSED BIT
```

```
/* ENABLE SERIOUS EXCEPTION MODE
```

```
/* PARITY SELECT (0=ODD, 1=EVEN)
```

```
/* RECORDING FORMAT
```

```
/* RECORDING DENSITY AND METHOD
```

```
/* SPARE UNUSED BIT
```

```
/* LOG SOFT (TU78) ERRORS (0=NO, 1=YES)
```

```
/* TOGGLE TO REVERSE LOGSOFT STATE BIT
```

```
/* AT BEGINNING OF TAPE
```

```
/* AT END OF FILE
```

```
/* AT END OF TAPE
```

```
/* TAPE IS HARDWARE WRITELOCKED
```

```
/* TAPE POSITION LOST
```

```
/* DRIVE SUPPORTS NRZI (800 BPI)
```

```
/* DRIVE SUPPORTS PE (1600 BPI)
```

```
/* DRIVE SUPPORTS GCR (6250 BPI)
```

```
/* TAPE SPEED
```

```
/*
```

```
/* RECORDING FORMAT DEFINITIONS
```

```
/*
```

```
constant 'DEFAULT' equals 0 prefix MT tag $K;
```

```
constant NORMAL11 equals 12 prefix MT tag $K;
```

```
constant CORDMP11 equals 13 prefix MT tag $K;
```

```
constant NORMAL15 equals 14 prefix MT tag $K;
```

```
/*
```

```
/* DEFAULT FORMAT
```

```
/* PDP-11 NORMAL
```

```
/* PDP-11 CORE DUMP
```

```
/* PDP-15 NORMAL
```

```
/*
```

```
/* RECORDING DENSITY AND METHOD DEFINITIONS
```

```
/*
```

```
/*
```

```
  DEFAULT,0
```

```
/* DEFAULT DENSITY (SAME AS ABOVE)
```

```
constant NRZI_800 equals 3 prefix MT tag $K;
```

```
constant PE_1600 equals 4 prefix MT tag $K;
```

```
constant GCR_6250 equals 5 prefix MT tag $K;
```

```
/* NRZI 800 BPI
```

```
/* PE 1600 BPI
```

```
/* GCR 6250 BPI
```

```
/*
```

```
/* TAPE SPEED VALUE DEFINITIONS
```

```
/*
```

```
constant SPEED_DEF equals 0 prefix MT tag $K;
```

```
constant SPEED_25 equals 25 prefix MT tag $K;
```

```
constant SPEED_75 equals 75 prefix MT tag $K;
```

```
/* DEFAULT SPEED
```

```
/* 25 IPS
```

```
/* 75 IPS
```

```
end MTDEF;
```

STARDEFMP.SDL:1

16-SEP-1984 16:46:59.44 Page 12

end\_module SMTDEF;

STA

mod  
/\*+  
/\*  
/\*-

con  
con  
con  
con  
con  
con  
con  
con  
con  
con

con  
con  
con  
con  
con  
con  
con

end



```
module $NSARECDEF;
```

```
/*+  
/* Security Auditing record definitions  
/*-
```

```
constant REC_MAXLENGTH equals 1024 tag C prefix NSAS; /* Maximum record size  
constant REC_MAXLENGTH equals 1024 tag K prefix NSAS; /* Maximum record size  
constant REC_MAXLENGTH equals 1024 tag S prefix NSAS; /* Maximum record size
```

```
/*+  
/* Audit record type definitions  
/*-
```

```
constant (RECTYP_FIL,          /* File access  
RECTYP_SYSUAF,              /* System UAF  
RECTYP_NETUAF,              /* Network UAF  
RECTYP_LOGB,                /* Login breakin detection  
RECTYP_LOGI,                /* Successful login  
RECTYP_LOGF,                /* Login failure  
RECTYP_LOGO,                /* Logout  
RECTYP_VOL)                 /* Volume operations  
equals T increment 1 counter #TYPNUM prefix NSAS;
```

```
constant RECTYPNUM equals #TYPNUM prefix NSAS;
```

```
/*+  
/* Audit record subtype and ID definitions  
/*-
```

```
/* File access
```

```
constant (RECTYP_FIL_SUCC,      /* Successful file access  
RECTYP_FIL_FAIL)              /* File access failure  
equals T increment 1 counter #SUBTYPNUM prefix NSAS;
```

```
constant RECTYPNUM_FIL equals #SUBTYPNUM prefix NSAS;
```

```
constant RECID_FIL_SUCC equals NSASK_RECTYP_FIL+(65536*NSASK_RECTYP_FIL_SUCC) prefix NSAS;  
constant RECID_FIL_FAIL equals NSASK_RECTYP_FIL+(65536*NSASK_RECTYP_FIL_FAIL) prefix NSAS;
```

```
/* System UAF
```

```
constant (RECTYP_SYSUAF_ADD,    /* System UAF record addition  
RECTYP_SYSUAF_DEL,            /* System UAF record deletion  
RECTYP_SYSUAF_MOD,            /* System UAF record modification  
RECTYP_SYSUAF_COP,            /* System UAF record copied  
RECTYP_SYSUAF_REN)            /* System UAF record renamed  
equals T increment 1 counter #SUBTYPNUM prefix NSAS;
```

```
constant RECTYPNUM_SYSUAF equals #SUBTYPNUM prefix NSAS;
```

```
constant RECID_SYSUAF_ADD equals NSASK_RECTYP_SYSUAF+(65536*NSASK_RECTYP_SYSUAF_ADD) prefix NSAS;
```

```
constant RECID_SYSUAF_DEL equals NSASK_RECTYP_SYSUAF+(65536*NSASK_RECTYP_SYSUAF_DEL) prefix NSAS;
constant RECID_SYSUAF_MOD equals NSASK_RECTYP_SYSUAF+(65536*NSASK_RECTYP_SYSUAF_MOD) prefix NSAS;
constant RECID_SYSUAF_COP equals NSASK_RECTYP_SYSUAF+(65536*NSASK_RECTYP_SYSUAF_COP) prefix NSAS;
constant RECID_SYSUAF_REN equals NSASK_RECTYP_SYSUAF+(65536*NSASK_RECTYP_SYSUAF_REN) prefix NSAS;
```

## /\* Network UAF

```
constant (RECTYP_NETUAF_ADD,      /* Network UAF record addition
      RECTYP_NETUAF_DEL,          /* Network UAF record deletion
      RECTYP_NETUAF_MOD)          /* Network UAF record modification
      equals T increment 1 counter #SUBTYPNUM prefix NSAS;
```

```
constant RECTYPNUM_NETUAF equals #SUBTYPNUM prefix NSAS;
```

```
constant RECID_NETUAF_ADD equals NSASK_RECTYP_NETUAF+(65536*NSASK_RECTYP_NETUAF_ADD) prefix NSAS;
constant RECID_NETUAF_DEL equals NSASK_RECTYP_NETUAF+(65536*NSASK_RECTYP_NETUAF_DEL) prefix NSAS;
constant RECID_NETUAF_MOD equals NSASK_RECTYP_NETUAF+(65536*NSASK_RECTYP_NETUAF_MOD) prefix NSAS;
```

## /\* Login breakin detection

```
constant (RECTYP_LOGB_DIA,      /* Dialup interactive breakin detection
      RECTYP_LOGB_LOC,          /* Local interactive breakin detection
      RECTYP_LOGB_REM,          /* Remote interactive breakin detection
      RECTYP_LOGB_NET,          /* Network breakin detection
      RECTYP_LOGB_DET)          /* Detached process breakin detection
      equals T increment 1 counter #SUBTYPNUM prefix NSAS;
```

```
constant RECTYPNUM_LOGB equals #SUBTYPNUM prefix NSAS;
```

```
constant RECID_LOGB_DIA equals NSASK_RECTYP_LOGB+(65536*NSASK_RECTYP_LOGB_DIA) prefix NSAS;
constant RECID_LOGB_LOC equals NSASK_RECTYP_LOGB+(65536*NSASK_RECTYP_LOGB_LOC) prefix NSAS;
constant RECID_LOGB_REM equals NSASK_RECTYP_LOGB+(65536*NSASK_RECTYP_LOGB_REM) prefix NSAS;
constant RECID_LOGB_NET equals NSASK_RECTYP_LOGB+(65536*NSASK_RECTYP_LOGB_NET) prefix NSAS;
constant RECID_LOGB_DET equals NSASK_RECTYP_LOGB+(65536*NSASK_RECTYP_LOGB_DET) prefix NSAS;
```

## /\* Successful login

```
constant (RECTYP_LOGI_BAT,      /* Batch process login
      RECTYP_LOGI_DIA,          /* Dialup interactive login
      RECTYP_LOGI_LOC,          /* Local interactive login
      RECTYP_LOGI_REM,          /* Remote interactive login
      RECTYP_LOGI_NET,          /* Network login
      RECTYP_LOGI_SUB,          /* Subprocess login
      RECTYP_LOGI_DET)          /* Detached process login
      equals T increment 1 counter #SUBTYPNUM prefix NSAS;
```

```
constant RECTYPNUM_LOGI equals #SUBTYPNUM prefix NSAS;
```

```
constant RECID_LOGI_BAT equals NSASK_RECTYP_LOGI+(65536*NSASK_RECTYP_LOGI_BAT) prefix NSAS;
constant RECID_LOGI_DIA equals NSASK_RECTYP_LOGI+(65536*NSASK_RECTYP_LOGI_DIA) prefix NSAS;
constant RECID_LOGI_LOC equals NSASK_RECTYP_LOGI+(65536*NSASK_RECTYP_LOGI_LOC) prefix NSAS;
constant RECID_LOGI_REM equals NSASK_RECTYP_LOGI+(65536*NSASK_RECTYP_LOGI_REM) prefix NSAS;
constant RECID_LOGI_NET equals NSASK_RECTYP_LOGI+(65536*NSASK_RECTYP_LOGI_NET) prefix NSAS;
```

```
constant RECID_LOGI_SUB equals NSASK_RECTYP_LOGI+(65536*NSASK_RECTYP_LOGI_SUB) prefix NSAS;
constant RECID_LOGI_DET equals NSASK_RECTYP_LOGI+(65536*NSASK_RECTYP_LOGI_DET) prefix NSAS;
```

## /\* Login failure

```
constant (RECTYP_LOGF_BAT,      /* Batch process login failure
RECTYP_LOGF_DIA,      /* Dialup interactive login failure
RECTYP_LOGF_LOC,      /* Local interactive login failure
RECTYP_LOGF_REM,      /* Remote interactive login failure
RECTYP_LOGF_NET,      /* Network login failure
RECTYP_LOGF_SUB,      /* Subprocess login failure
RECTYP_LOGF_DET)      /* Detached process login failure
equals T increment 1 counter #SUBTYPNUM prefix NSAS;
```

```
constant RECTYPNUM_LOGF equals #SUBTYPNUM prefix NSAS;
```

```
constant RECID_LOGF_BAT equals NSASK_RECTYP_LOGF+(65536*NSASK_RECTYP_LOGF_BAT) prefix NSAS;
constant RECID_LOGF_DIA equals NSASK_RECTYP_LOGF+(65536*NSASK_RECTYP_LOGF_DIA) prefix NSAS;
constant RECID_LOGF_LOC equals NSASK_RECTYP_LOGF+(65536*NSASK_RECTYP_LOGF_LOC) prefix NSAS;
constant RECID_LOGF_REM equals NSASK_RECTYP_LOGF+(65536*NSASK_RECTYP_LOGF_REM) prefix NSAS;
constant RECID_LOGF_NET equals NSASK_RECTYP_LOGF+(65536*NSASK_RECTYP_LOGF_NET) prefix NSAS;
constant RECID_LOGF_SUB equals NSASK_RECTYP_LOGF+(65536*NSASK_RECTYP_LOGF_SUB) prefix NSAS;
constant RECID_LOGF_DET equals NSASK_RECTYP_LOGF+(65536*NSASK_RECTYP_LOGF_DET) prefix NSAS;
```

## /\* Logout

```
constant (RECTYP_LOGO_BAT,      /* Batch process logout
RECTYP_LOGO_DIA,      /* Dialup interactive logout
RECTYP_LOGO_LOC,      /* Local interactive logout
RECTYP_LOGO_REM,      /* Remote interactive logout
RECTYP_LOGO_NET,      /* Network logout
RECTYP_LOGO_SUB,      /* Subprocess logout
RECTYP_LOGO_DET)      /* Detached process logout
equals T increment 1 counter #SUBTYPNUM prefix NSAS;
```

```
constant RECTYPNUM_LOGO equals #SUBTYPNUM prefix NSAS;
```

```
constant RECID_LOGO_BAT equals NSASK_RECTYP_LOGO+(65536*NSASK_RECTYP_LOGO_BAT) prefix NSAS;
constant RECID_LOGO_DIA equals NSASK_RECTYP_LOGO+(65536*NSASK_RECTYP_LOGO_DIA) prefix NSAS;
constant RECID_LOGO_LOC equals NSASK_RECTYP_LOGO+(65536*NSASK_RECTYP_LOGO_LOC) prefix NSAS;
constant RECID_LOGO_REM equals NSASK_RECTYP_LOGO+(65536*NSASK_RECTYP_LOGO_REM) prefix NSAS;
constant RECID_LOGO_NET equals NSASK_RECTYP_LOGO+(65536*NSASK_RECTYP_LOGO_NET) prefix NSAS;
constant RECID_LOGO_SUB equals NSASK_RECTYP_LOGO+(65536*NSASK_RECTYP_LOGO_SUB) prefix NSAS;
constant RECID_LOGO_DET equals NSASK_RECTYP_LOGO+(65536*NSASK_RECTYP_LOGO_DET) prefix NSAS;
```

## /\* Volume operations

```
constant (RECTYP_VOL_MOU,      /* Volume mounts
RECTYP_VOL_DMOU)      /* Volume dismounts
equals T increment 1 counter #SUBTYPNUM prefix NSAS;
```

```
constant RECTYPNUM_VOL equals #SUBTYPNUM prefix NSAS;
```



```

constant RECID_VOL_MOU equals NSASK_RECTYP_VOL+(65536*NSASK_RECTYP_VOL_MOU) prefix NSA$;
constant RECID_VOL_DMOU equals NSASK_RECTYP_VOL+(65536*NSASK_RECTYP_VOL_DMOU) prefix NSA$;

```

```

/*+
/* Record header offset definitions
/*-

```

```

aggregate NSARECHDRDEF structure prefix NSA$;

```

```

    REC_ID_OVERLAY union fill;
        REC_ID longword unsigned; /* Record identification longword
        REC_ID_FIELDS structure fill;
            REC_TYPE word unsigned; /* Record type
            REC_SUBTYPE word unsigned; /* Record subtype
        end REC_ID_FIELDS;
    end REC_ID_OVERLAY;
    REC_SEQNUM byte unsigned; /* This records sequence number
    REC_SEQLAST byte unsigned; /* Last records sequence number
    REC_FLAGS_OVERLAY union fill;
        REC_FLAGS byte unsigned; /* Record flags byte
        REC_FLAGS_BITS structure fill;
            REC_FLAGS_PKTCON bitfield length 1 mask; /* Last packet in record is
                                                    /* continued in next record
    end REC_FLAGS_BITS;
    end REC_FLAGS_OVERLAY;
    REC_PKTNUM byte unsigned; /* Number of data packets in record
    REC_PKTOFF word unsigned; /* Offset to first packet
    REC_PKTHDRSZ word unsigned; /* Data packet header size
    REC_EPID longword unsigned; /* Extended PID
    REC_TIME quadword unsigned; /* Event time (EXESGQ_SYSTIME)
    REC_CLUSNAM character length 16; /* Cluster node name
    REC_PROCNAM character length 16; /* Process name
    REC_USERNAME character length 12; /* username
    REC_ACCTNAM character length 8; /* Account name

```

```

    constant RECHDR_LENGTH equals . tag C;
    constant RECHDR_LENGTH equals . tag K;

```

```

end NSARECHDRDEF;

```

```

/*+
/* Data packet type definitions
/*-

```

```

constant (PKTTYP_IMGNAM, /* Image name packet
    PKTTYP_FACMOD, /* File access mode
    PKTTYP_PRIVUSED, /* Privilege used to access file
    PKTTYP_FILNAM, /* File name
    PKTTYP_DEVNAM, /* Device name
    PKTTYP_LOGNAM, /* Logical name
    PKTTYP_VOLNAM, /* Volume name
    PKTTYP_VOLSNAM, /* Volume set name
    PKTTYP_NODENAM, /* Node name
    PKTTYP_USERNAME, /* User name

```

```

PKTTYP_PASSWORD,      /* Password
PKTTYP_UIC,            /* User identification code
PKTTYP_VOLPRO,         /* Volume protection
PKTTYP_MOUFLG,         /* Mount flags
PKTTYP_DMOUFLG,        /* Dismount flags
PKTTYP_NODEID,         /* Node ID
PKTTYP_EPID,           /* Extended PID
PKTTYP_SYSUAFF,        /* System UAF record fields
PKTTYP_STATUS)         /* Status longword
equals T increment 1 counter #PKTTYPNUM prefix NSA$;

```

```
constant PKTTYPNUM equals #PKTTYPNUM prefix NSA$;
```

```

/*+
/* Data packet offset definitions
/*-

```

```
aggregate NSAPKTDEF structure origin PKT_DATA prefix NSA$;
```

```

#PKTHDRBEG = .;
PKT_TYPE word unsigned;      /* Packet data type
PKT_SIZE word unsigned;      /* Packet size
PKT_DATA character length 0;

```

```

constant PKTHDR_LENGTH equals .-#PKTHDRBEG tag C;
constant PKTHDR_LENGTH equals .-#PKTHDRBEG tag K;

```

```

PKT_DATA OVERLAY union fill;
  PKT_IMGNAME character length 44; /* Image name
  PKT_FACMOD longword unsigned;    /* File access mode
  PKT_PRIVUSED longword unsigned;  /* Privilege used to access file
  PKT_FILNAME character length 44; /* File name
  PKT_DEVDNAME character length 15; /* Device name
  PKT_LOGNAME character length 25; /* Logical name
  PKT_VOLNAME character length 12; /* Volume name
  PKT_VOLSNAME character length 12; /* Volume set name
  PKT_NODENAME character length 6; /* Node name
  PKT_USERNAME character length 12; /* User name
  PKT_PASSWORD character length 31; /* Password
  PKT_UIC longword unsigned;       /* Volume UIC
  PKT_VOLPRO word unsigned;        /* Volume protection
  PKT_MOUFLG longword unsigned;    /* Mount flags
  PKT_DMOUFLG word unsigned;       /* Dismount flags
  PKT_NODEID quadword unsigned;    /* Node ID
  PKT_EPID longword unsigned;      /* Extended PID
  PKT_SYSUAFF quadword unsigned;   /* System UAF record fields
  PKT_STATUS longword unsigned;    /* Status longword

```

```

end PKT_DATA_OVERLAY;
end NSAPKTDEF;

```

```
end_module $NSARECDEF;
```

```
module $OPRDEF;
```

```
/*  
/* OPERATOR COMMUNICATIONS MESSAGE TYPES AND VALUES  
/*-
```

```
constant(  
    TERMENABL  
    , LOGINIT  
    , OPRQST  
    , OPREPLY  
    ) equals 1 increment 1 prefix OPR tag $;
```

```
aggregate OPRDEF union prefix OPR$;  
    OPRDEF BITS structure fill;  
        CENTRAL bitfield mask;  
        PRINTER bitfield mask;  
        TAPES bitfield mask;  
        DISKS bitfield mask;  
        DEVICES bitfield mask;  
        USER1 bitfield mask length 12;  
        USER2 bitfield mask;  
        USER3 bitfield mask;  
        USER4 bitfield mask;  
        USER5 bitfield mask;  
        USER6 bitfield mask;  
        USER7 bitfield mask;  
        USER8 bitfield mask;  
        USER9 bitfield mask;  
        USER10 bitfield mask;  
        USER11 bitfield mask;  
        USER12 bitfield mask;  
    end OPRDEF_BITS;  
end OPRDEF;  
end_module $OPRDEF;
```

```
/* OPERATOR MESSAGE TYPES
```

```
/* ENABLE TERMINAL  
/* INITIALIZE THE LOG  
/* OPERATOR REQUEST  
/* OPERATOR REPLY
```

```
/*
```



```
module $PCCDEF;
/*+
/* PRINTER/TERMINAL CARRIAGE CONTROL SPECIFIERS
/*--

aggregate PCCDEF structure prefix PCC$:
  FORTRAN word unsigned;          /* FORTRAN FIELD
  "PREFIX" byte unsigned;         /* PREFIX FIELD
  POSTFIX OVERLAY union fill;
    POSTFIX byte unsigned;        /* POSTFIX FIELD
    POSTFIX BITS0 structure fill;
      CHAR bitfield mask length 5; /* CHARACTER FIELD
      FILL 1 bitfield fill prefix PCCDEF tag $$;
      EIGHTBIT bitfield mask;     /* EIGHTBIT CHARACTER SET
      SINGLE bitfield mask;       /* SINGLE CHARACTER
    end POSTFIX BITS0;
    POSTFIX BITS1 structure fill;
      LINECNT bitfield mask length 7; /* LINE COUNT FOR NEWLINES
    end POSTFIX BITS1;

    /* FORTRAN CONSTANTS
    constant FTN_SINGLE equals 32 prefix PCC tag $; /* SINGLE SPACE
    constant FTN_DOUBLE equals 48 prefix PCC tag $; /* DOUBLE SPACE
    constant FTN_PAGE equals 49 prefix PCC tag $; /* PAGE SPACE
    constant FTN_OVRPRT equals 43 prefix PCC tag $; /* OVER PRINT
    constant FTN_PROMPT equals 36 prefix PCC tag $; /* PROMPT
  end POSTFIX_OVERLAY;
end PCCDEF;

end_module $PCCDEF;
```

```
module $PLVDEF;
```

```
/*  
/* PRIVILEGED LIBRARY VECTOR DEFINITION  
/*-
```

```
aggregate PLVDEF structure prefix PLV$;
```

```
TYPE longword unsigned;
```

```
constant(  
  TYP-CMOD
```

```
  TYP-MSG  
  ) equals 1 increment 1 prefix PLV tag $C;
```

```
VERSION longword unsigned;
```

```
KERNEL OVERLAY union fill;
```

```
  KERNEL longword unsigned;
```

```
  MSGDSP longword unsigned;
```

```
end KERNEL_OVERLAY;
```

```
EXEC longword unsigned;
```

```
USRUNDWN longword unsigned;
```

```
FILL 1 longword fill prefix PLVDEF tag $$;
```

```
RMS longword unsigned;
```

```
CHECK longword unsigned;
```

```
end PLVDEF;
```

```
end_module $PLVDEF;
```

```
/*TYPE CODE FOR VECTOR FORMAT
```

```
/*TYPE CODES FOR PRIVILEGE VECTORS
```

```
/*CHANGE MODE VECTOR TYPE
```

```
/*MESSAGE VECTOR TYPE
```

```
/*SYSTEM VERSION NUMBER
```

```
/*SELF-REL PTR TO KERNEL MODE DISPATCHER
```

```
/*SELF-REL PTR TO MESSAGE DISPATCHER
```

```
/*SELF-REL PTR TO EXEC MODE DISPATCHER
```

```
/*SELF-REL PTR TO USER RUNDOWN SERVICE
```

```
/*UNUSED, RESERVED FOR FUTURE USE
```

```
/*SELF-REL PTR TO RMS SERVICES DISPATCHER
```

```
/*LONGWORD USED TO CHECK VIRTUAL ADDRESS
```

```
/*LOCATION OF VECTOR
```

module \$PQLDEF;

/\*+

/\* PROCESS QUOTA LIST CODES

/\*-

constant(

LISTEND

, ASTLM

, BIOLM

, BYTLM

, CPULM

, DIOLM

, FILLM

, PGFLQUOTA

, PRCLM

, TQELM

, WSQUOTA

, WSDEFAULT

, ENQLM

, WSEXTENT

, JTQUOTA

, 'LENGTH'

) equals 0 increment 1 prefix PQL tag \$;

end\_module \$PQLDEF;

/\*LIST END CODE (MUST BE FIRST)

/\*AST LIMIT

/\*BUFFERED I/O LIMIT

/\*BYTE LIMIT FOR BUFFERED I/O

/\*CPU TIME LIMIT

/\*DIRECT I/O LIMIT

/\*OPEN FILE LIMIT

/\*PAGING FILE QUOTA

/\*SUB-PROCESS LIMIT

/\*TIMER QUEUE ENTRY LIMIT

/\*WORKING SET QUOTA

/\*WORKING SET DEFAULT

/\*ENQUEUE LIMIT

/\*WORKING SET EXTENT LIMIT

/\*JOB-WIDE LOGICAL NAME TABLE CREATION QUOTA

/\*NUMBER OF QUOTAS (MUST BE LAST)

module \$PRCDEF;

/\*+

/\* \$CREPRC STATUS FLAGS AND ITEM CODES

/\*-

aggregate PRCDEF union prefix PRC\$;

PRCDEF BITS structure fill;

SSRWAIT bitfield mask;

SSFEXCU bitfield mask;

PSWAPM bitfield mask;

NOACNT bitfield mask;

BATCH bitfield mask;

HIBER bitfield mask;

NOUAF bitfield mask;

NETWRK bitfield mask;

DISAWS bitfield mask;

DETACH bitfield mask;

INTER bitfield mask;

IMGDMP bitfield mask;

CLISPEC bitfield mask;

NOPASSWORD bitfield mask;

end PRCDEF BITS;

PRCDEF OBSOLETE structure fill;

FILL 0 bitfield length 6 fill;

LOGIN bitfield mask;

end PRCDEF\_OBSOLETE;

end PRCDEF;

/\*+

/\*

/\* Create Process Item List Data Identifier Definitions

/\*

/\* \*\*\*\*\* NOTE \*\*\*\*\*

/\*

/\* New items must always be added at the END of the list so that

/\* users will not have to relink or reassemble.

/\*

/\*-

constant(

LISTEND

, PGFLCHAR

, PGFLINDEX

, INPUT\_ATT

, OUTPUT\_ATT

, ERROR\_ATT

) equals 0 increment 1 prefix PRC tag \$;

end\_module \$PRCDEF;

/\* RESOURCE WAIT DISABLE

/\* SYSTEM SERVICE FAIL EXCEPTION MODE

/\* PROCESS SWAP MODE

/\* ACCOUNTING MESSAGE DISABLE

/\* BATCH INDICATOR

/\* HIBERNATE BEFORE CALLING INITIAL IMAGE

/\* BYPASS LOGIN VERIFICATION FOR DETACHED PROC.

/\* NETWORK INDICATOR

/\* DISABLE WORKING SET ADJUST

/\* DETACHED PROCESS

/\* INTERACTIVE INDICATOR

/\* IMAGE DUMP REQUESTED

/\* PASS CLI SPECIFICATIONS

/\* DON'T PROMPT FOR USERNAME AND PASSWORD

/\* BYPASS LOGIN VERIFICATION FOR DETACHED PROC.

/\* End of list (must be first code)

/\* Page file characteristics

/\* Page file index

/\* SYSSINPUT attributes

/\* SYSSOUTPUT attributes

/\* SYSSERROR attributes

STA

mod

{+}

{+}

{+}

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

con

agg



```
module $PRVDEF;
```

```
/*  
/* PRIVILEGE BIT DEFINITIONS  
/*  
/* Note that any privileges added here must also be reflected in the  
/* modules [VMSLIB.SRC]SETPRIV.MAR, [CLIUTL.SRC]SHOWPROC.B32,  
/* [RTL.SRC]LIBLEXICA.B32, and  
/* [CLD.SRC]DCLINT.CLD, MCRINT.CLD, MCRSET.CLD, RUN.CLD, and SET.CLD  
/* to completely add the new privilege.  
/*  
/*-
```

```
aggregate PRVDEF union prefix PRVS;
```

```
PRVDEF BITSO structure fill;  
CMKRNL bitfield mask;  
CMEXEC bitfield mask;  
/* ***** THE PRECEEDING TWO BITS MUST BE ADJACENT  
/* ***** THE FOLLOWING TWO BITS MUST BE ADJACENT  
SYSNAM bitfield mask;  
GRPNAM bitfield mask;  
/* ***** THE PRECEEDING TWO BITS MUST BE ADJACENT  
ALLSPOOL bitfield mask;  
DETACH bitfield mask;  
DIAGNOSE bitfield mask;  
LOG IO bitfield mask;  
GROOP bitfield mask;  
NOACNT bitfield mask;  
PRMCEB bitfield mask;  
PRMMBX bitfield mask;  
PSUAPH bitfield mask;  
SETPRI bitfield mask;  
SETPRV bitfield mask;  
TMPMBX bitfield mask;  
WORLD bitfield mask;  
MOUNT bitfield mask;  
OPER bitfield mask;  
EXQUOTA bitfield mask;  
NETMBX bitfield mask;  
VOLPRO bitfield mask;  
PHY IO bitfield mask;  
BUGCHK bitfield mask;  
PRMGBL bitfield mask;  
SYSGBL bitfield mask;  
PFNMAP bitfield mask;  
SHMEM bitfield mask;  
SYSPRV bitfield mask;  
BYPASS bitfield mask;  
SYSLCK bitfield mask;  
SHARE bitfield mask;  
{  
The following bits are in the second longword,  
and thus, cannot have prv$m_ symbols...  
UPGRADE bitfield;  
DOWNGRADE bitfield;  
GRPPRV bitfield;
```

```
/* MAY CHANGE MODE TO KERNEL  
/* MAY CHANGE MODE TO EXEC
```

```
/* MAY INSERT IN SYSTEM LOGICAL NAME TABLE  
/* MAY INSERT IN GROUP LOGICAL NAME TABLE
```

```
/*MAY ALLOCATE SPOOLED DEVICE  
/* MAY CREATE DETACHED PROCESSES  
/* MAY DIAGNOSE DEVICES  
/* MAY DO LOGICAL I/O  
/* MAY AFFECT OTHER PROCESSES IN SAME GROUP  
/* MAY SUPPRESS ACCOUNTING MESSAGE  
/* MAY CREATE PERMANENT COMMON EVENT CLUSTERS  
/* MAY CREATE PERMANENT MAILBOX  
/* MAY CHANGE PROCESS SWAP MODE  
/* MAY SET ANY PRIORITY VALUE  
/* MAY SET ANY PRIVILEGE BITS  
/* MAY CREATE TEMPORARY MAILBOX  
/* MAY AFFECT OTHER PROCESSES IN THE WORLD  
/* MAY EXECUTE MOUNT ACP FUNCTIONS  
/* OPERATOR PRIVILEGE  
/* MAY EXCEED QUOTAS  
/* MAY CREATE NETWORK DEVICE  
/* MAY OVERRIDE VOLUME PROTECTION  
/* MAY DO PHYSICAL I/O  
/* MAY MAKE BUG CHECK ERROR LOG ENTRIES  
/* MAY CREATE PERMANENT GLOBAL SECTIONS  
/* MAY CREATE SYSTEM WIDE GLOBAL SECTIONS  
/* MAY MAP TO SECTION BY PFN  
/* MAY ALLOCATE STRUCTURES IN SHARED MEMORY  
/* ELIGIBLE FOR SYSTEM PROTECTION FIELD  
/* MAY BYPASS UIC BASED PROTECTION  
/* MAY CREATE SYSTEM WIDE LOCKS  
/* MAY ASSIGN CHANNEL TO NON-SHARED DEVICE
```

```
/* May upgrade classification  
/* May downgrade classification  
/* Group access via system protection field
```

```
      READALL bitfield;          /* Read access to everything
      TMPJNL bitfield;          /* May create temporary journals
      PRMJNL bitfield;          /* May create permanent journals
      SECURITY bitfield;        /* May perform security functions
end PRVDEF_BITS0;

      PRVDEF_BITS1 structure fill;
      FICL_1 bitfield length 9 fill prefix PRVDEF tag $$; /* SKIP 9
      ACNT bitfield mask;      /* MAY SUPPRESS ACCOUNTING MESSAGES (NOACNT)
      FILL_2 bitfield length 3 fill prefix PRVDEF tag $$; /* SKIP 3
      ALTPRI bitfield mask;    /* MAY SET ANY PRIORITY VALE (SETPRI)
end PRVDEF_BITS1;
end PRVDEF;
end_module $PRVDEF;
```

module \$PRTDEF;

/\*+

/\* PROTECTION FIELD DEFINITIONS

/\*-

constant NA	equals	(XB0000)	prefix	PRT	tag	\$C;	/* NO ACCESS
constant KR	equals	(XB0011)	prefix	PRT	tag	\$C;	/* KERNEL READ ONLY
constant KW	equals	(XB0010)	prefix	PRT	tag	\$C;	/* KERNEL WRITE
constant ER	equals	(XB0111)	prefix	PRT	tag	\$C;	/* EXEC READ ONLY
constant EW	equals	(XB0101)	prefix	PRT	tag	\$C;	/* EXEC WRITE
constant SR	equals	(XB1011)	prefix	PRT	tag	\$C;	/* SUPER READ ONLY
constant SW	equals	(XB1000)	prefix	PRT	tag	\$C;	/* SUPER WRITE
constant UR	equals	(XB1111)	prefix	PRT	tag	\$C;	/* USER READ ONLY
constant UW	equals	(XB0100)	prefix	PRT	tag	\$C;	/* USER WRITE
constant ERKW	equals	(XB0110)	prefix	PRT	tag	\$C;	/* EXEC READ KERNEL WRITE
constant SRKW	equals	(XB1010)	prefix	PRT	tag	\$C;	/* SUPER READ KERNEL WRITE
constant SREW	equals	(XB1001)	prefix	PRT	tag	\$C;	/* SUPER READ EXEC WRITE
constant URKW	equals	(XB1110)	prefix	PRT	tag	\$C;	/* USER READ KERNEL WRITE
constant UREW	equals	(XB1101)	prefix	PRT	tag	\$C;	/* USER READ EXEC WRITE
constant URSW	equals	(XB1100)	prefix	PRT	tag	\$C;	/* USER READ SUPER WRITE
constant RESERVED	equals	1	prefix	PRT	tag	\$C;	/* RESERVED

end\_module \$PRTDEF;

module \$PRDEF;

```

/*
/* PROCESSOR REGISTER DEFINITIONS
/*-

```

```

constant KSP      equals 0  prefix PR tag $;
constant ESP      equals 1  prefix PR tag $;
constant SSP      equals 2  prefix PR tag $;
constant USP      equals 3  prefix PR tag $;
constant ISP      equals 4  prefix PR tag $;
constant POBR     equals 8  prefix PR tag $;
constant POLR     equals 9  prefix PR tag $;
constant P1BR     equals 10 prefix PR tag $;
constant P1LR     equals 11 prefix PR tag $;
constant SBR      equals 12 prefix PR tag $;
constant SLR      equals 13 prefix PR tag $;
constant PCBB     equals 16 prefix PR tag $;
constant SCBB     equals 17 prefix PR tag $;
constant IPL      equals 18 prefix PR tag $;
constant ASTLVL   equals 19 prefix PR tag $;
constant SIRR     equals 20 prefix PR tag $;
constant SISR     equals 21 prefix PR tag $;
constant ICCS     equals 24 prefix PR tag $;
constant RXCS     equals 32 prefix PR tag $;
constant RXDB     equals 33 prefix PR tag $;
constant TXCS     equals 34 prefix PR tag $;
constant TXDB     equals 35 prefix PR tag $;
constant MAPEN    equals 56 prefix PR tag $;
constant TBIA     equals 57 prefix PR tag $;
constant TBIS     equals 58 prefix PR tag $;
constant SID      equals 62 prefix PR tag $;
constant TBCHK    equals 63 prefix PR tag $;

```

```

/*KERNEL STACK POINTER
/*EXECUTIVE STACK POINTER
/*SUPERVISOR STACK POINTER
/*USER STACK POINTER
/*INTERRUPT STACK POINTER
/*PO BASE REGISTER
/*PO LIMIT REGISTER
/*P1 BASE REGISTER
/*P1 LIMIT REGISTER
/*SYSTEM BASE REGISTER
/*SYSTEM LIMIT REGISTER
/*PROCESS CONTROL BLOCK BASE
/*SYSTEM CONTROL BLOCK BASE
/*INTERRUPT PRIORITY LEVEL REGISTER
/*AST LEVEL REGISTER
/*SOFTWARE INTERRUPT REQUEST REGISTER
/*SOFTWARE INTERRUPT SUMMARY REGISTER
/* INTERVAL CLOCK CONTROL STATUS REGISTER
/* CONSOLE RECIEVER CONTROL STATUS REGISTER
/* CONSOLE RECEIVER DATA BUFFER REGISTER
/* CONSOLE TRANSMIT CONTROL STATUS REGISTER
/* CONSOLE TRANSMIT DATA BUFFER REGISTER
/* MAPPING ENABLE REGISTER
/* TRANSLATION BUFFER INVALIDATE ALL
/* TRANSLATION BUFFER INVALIDATE SINGLE
/* SYSTEM IDENTIFICATION REGISTER
/* TRANSLATION BUFFER VALID CHECK

```

```

aggregate PRDEF union prefix PR$:
  PRDEF_BITS structure fill:
    SID_SN bitfield length 12;
    SID_PL bitfield length 3;
    SID_ECO bitfield length 9;
    SID_TYPE bitfield length 8;
  end PRDEF_BITS;

```

```

/* SERIAL NUMBER FIELD
/* PLANT ID
/* ECO LEVEL
/* CPU TYPE CODE

```

```

constant TYP780   equals 1  prefix PR$_S tag ID:
constant TYP750   equals 2  prefix PR$_S tag ID:
constant TYP730   equals 3  prefix PR$_S tag ID:
constant TYP790   equals 4  prefix PR$_S tag ID:
constant TYP8SS   equals 5  prefix PR$_S tag ID:
constant TYP8NN   equals 6  prefix PR$_S tag ID:
constant TYPMAX   equals 8  prefix PR$_S tag ID:

constant TYPUV1   equals 7  prefix PR$_S tag ID:
constant TYPUV2   equals 8  prefix PR$_S tag ID:

constant WCSA     equals 44  prefix PR tag $;
constant WCSD     equals 45  prefix PR tag $;

```

```

/* SYSTEM ID REGISTER CPU TYPES
/* VAX 11/780
/* VAX 11/750
/* VAX 11/730
/* VAX 11/790
/* Scorpio for now
/* Nautilus for now
/* MAX LEGAL CPU TYPE
/* Micro-VAX cpus
/* Micro-VAX UV1
/* Micro-VAX UV2
/*VAX 11/780 IPR's:
/* WCS ADDRESS REGISTER
/* WCS DATA REGISTER

```



```

constant SBIFS      equals 48 prefix PR tag $: /* SBI FAULT STATUS REGISTER
constant SBIS       equals 49 prefix PR tag $: /* SBI SILO REGISTER
constant SBISC      equals 50 prefix PR tag $: /* SBI COMPARATOR REGISTER
constant SBIMT      equals 51 prefix PR tag $: /* SBI MAINTENANCE REGISTER
constant SBIER      equals 52 prefix PR tag $: /* SBI ERROR REGISTER
constant SBITA      equals 53 prefix PR tag $: /* SBI TIMEOUT ADDRESS REGISTER
constant SBIQC      equals 54 prefix PR tag $: /* SBI QUADWORD CLEAR REGISTER
/*END OF VAX 11/780-SPECIFIC IPR'S
/*VAX 11/750 AND 11/730 IPR'S:
constant CMIERR     equals 23 prefix PR tag $: /* CMI ERROR SUMMARY REGISTER
constant CSRS       equals 28 prefix PR tag $: /* CONSOLE BLK STORE RCV STATUS
constant CSRD       equals 29 prefix PR tag $: /* CONSOLE BLK STORE RCV DATA
constant CSTS       equals 30 prefix PR tag $: /* CONSOLE BLK STORE XMIT STATUS
constant CSTD       equals 31 prefix PR tag $: /* CONSOLE BLK STORE XMIT DATA
constant TBDR       equals 36 prefix PR tag $: /* TB DISABLE REGISTER
constant CADR       equals 37 prefix PR tag $: /* CACHE DISABLE REGISTER
constant MCESR      equals 38 prefix PR tag $: /* MACHINE CHECK ERROR SUMMARY REG
constant CAER       equals 39 prefix PR tag $: /* CACHE ERROR REGISTER
constant UBRESET    equals 55 prefix PR tag $: /* UNIBUS I/O RESET REGISTER
/*END OF 11/750 AND 11/730 IPR'S
/*VAX 11/790 PROCESSOR-SPECIFIC IPRS
constant PAMACC     equals 64 prefix PR tag $: /* PAMM ACCESS
constant PAMLOC     equals 65 prefix PR tag $: /* PAMM LOCATION
constant CSWP       equals 66 prefix PR tag $: /* CACHE SWEEP REGISTER
constant MDECC      equals 67 prefix PR tag $: /* MBOX DATA ECC REGISTER
constant MENA       equals 68 prefix PR tag $: /* MBOX ERROR ENABLE REGISTER
constant MDCTL      equals 69 prefix PR tag $: /* MBOX DATA CONTROL REGISTER
constant MCCTL      equals 70 prefix PR tag $: /* MBOX MCC CONTROL REGISTER
constant MERG       equals 71 prefix PR tag $: /* MBOX ERROR GENERATOR REGISTER
constant CRBT       equals 72 prefix PR tag $: /* CONSOLE REBOOT
constant DFI        equals 73 prefix PR tag $: /* DIAGNOSTIC FAULT INSERTION
constant EHSR       equals 74 prefix PR tag $: /* ERROR HANDLING STATUS REGISTER
constant ACCS790    equals 75 prefix PR tag $: /* ACCELERATOR STATUS REGISTER
constant STXCS      equals 76 prefix PR tag $: /* CONSOLE STORAGE CONTROL REG
constant STXDB      equals 77 prefix PR tag $: /* CONSOLE STORAGE DATA REGISTER
constant LSPA       equals 78 prefix PR tag $: /* SCRATCHPAD ADDRESS
constant RSPD       equals 79 prefix PR tag $: /* SCRATCHPAD DATA
/*END OF 11/790 PROCESSOR-SPECIFIC IPRS

```

end PRDEF;

end\_module \$PRDEF;

module \$PR730DEF;

```
{**
{* 11/730-Specific Processor Register Definitions
{*-
```

```
constant NICR equals 25 prefix PR730 tag $: /* INTERVAL CLOCK NEXT INTERVAL REGISTER
constant ICR equals 26 prefix PR730 tag $: /* INTERVAL CLOCK INTERVAL COUNT REGISTER
constant TODR equals 27 prefix PR730 tag $: /* TIME OF DAY REGISTER
constant ACCS equals 40 prefix PR730 tag $: /* ACCELERATOR CONTROL STATUS REGISTER
constant ACCR equals 41 prefix PR730 tag $: /* ACCELERATOR RESERVED
constant PME equals 61 prefix PR730 tag $: /* PERFORMANCE MONITOR ENABLE

constant CMIERR equals 23 prefix PR730 tag $: /* CMI ERROR SUMMARY REGISTER
constant CSRS equals 28 prefix PR730 tag $: /* CONSOLE BLK STORE RCV STATUS
constant CSRD equals 29 prefix PR730 tag $: /* CONSOLE BLK STORE RCV DATA
constant CSTS equals 30 prefix PR730 tag $: /* CONSOLE BLK STORE XMIT STATUS
constant CSTD equals 31 prefix PR730 tag $: /* CONSOLE BLK STORE XMIT DATA
constant TBDR equals 36 prefix PR730 tag $: /* TB DISABLE REGISTER
constant CADR equals 37 prefix PR730 tag $: /* CACHE DISABLE REGISTER
constant MCESR equals 38 prefix PR730 tag $: /* MACHINE CHECK ERROR SUMMARY REG
constant CAER equals 39 prefix PR730 tag $: /* CACHE ERROR REGISTER
constant UBRESET equals 55 prefix PR730 tag $: /* UNIBUS I/O RESET REGISTER
```

end\_module \$PR730DEF;

module \$PR750DEF;

(\*+  
(\* 11/750-Specific Processor Register Definitions  
(\*-

constant	NICR	equals	25	prefix	PR750	tag	\$:	/*	INTERVAL CLOCK NEXT INTERVAL REGISTER
constant	ICR	equals	26	prefix	PR750	tag	\$:	/*	INTERVAL CLOCK INTERVAL COUNT REGISTER
constant	TODR	equals	27	prefix	PR750	tag	\$:	/*	TIME OF DAY REGISTER
constant	ACCS	equals	40	prefix	PR750	tag	\$:	/*	ACCELERATOR CONTROL STATUS REGISTER
constant	ACCR	equals	41	prefix	PR750	tag	\$:	/*	ACCELERATOR RESERVED
constant	PME	equals	61	prefix	PR750	tag	\$:	/*	PERFORMANCE MONITOR ENABLE
constant	CMIERR	equals	23	prefix	PR750	tag	\$:	/*	CM: ERROR SUMMARY REGISTER
constant	CSRS	equals	28	prefix	PR750	tag	\$:	/*	CONSOLE BLK STORE RCV STATUS
constant	CSRD	equals	29	prefix	PR750	tag	\$:	/*	CONSOLE BLK STORE RCV DATA
constant	CSTS	equals	30	prefix	PR750	tag	\$:	/*	CONSOLE BLK STORE XMIT STATUS
constant	CSTD	equals	31	prefix	PR750	tag	\$:	/*	CONSOLE BLK STORE XMIT DATA
constant	TBDR	equals	36	prefix	PR750	tag	\$:	/*	TB DISABLE REGISTER
constant	CADR	equals	37	prefix	PR750	tag	\$:	/*	CACHE DISABLE REGISTER
constant	MCESR	equals	38	prefix	PR750	tag	\$:	/*	MACHINE CHECK ERROR SUMMARY REG
constant	CAER	equals	39	prefix	PR750	tag	\$:	/*	CACHE ERROR REGISTER
constant	UBRESET	equals	55	prefix	PR750	tag	\$:	/*	UNIBUS I/O RESET REGISTER

end\_module \$PR750DEF;

```
module $PR780DEF;
```

```
{**  
{* 11/780-Specific Processor Register Definitions  
{*-
```

```
constant NICR equals 25 prefix PR780 tag $: /* INTERVAL CLOCK NEXT INTERVAL REGISTER  
constant ICR equals 26 prefix PR780 tag $: /* INTERVAL CLOCK INTERVAL COUNT REGISTER  
constant TODR equals 27 prefix PR780 tag $: /* TIME OF DAY REGISTER  
constant ACCS equals 40 prefix PR780 tag $: /* ACCELERATOR CONTROL STATUS REGISTER  
constant ACCR equals 41 prefix PR780 tag $: /* ACCELERATOR RESERVED  
constant PME equals 61 prefix PR780 tag $: /* PERFORMANCE MONITOR ENABLE  
  
constant WCSA equals 44 prefix PR780 tag $: /* WCS ADDRESS REGISTER  
constant WCSD equals 45 prefix PR780 tag $: /* WCS DATA REGISTER  
constant SBIFS equals 48 prefix PR780 tag $: /* SBI FAULT STATUS REGISTER  
constant SBIS equals 49 prefix PR780 tag $: /* SBI SILO REGISTER  
constant SBISC equals 50 prefix PR780 tag $: /* SBI COMPARATOR REGISTER  
constant SBIMT equals 51 prefix PR780 tag $: /* SBI MAINTENANCE REGISTER  
constant SBIER equals 52 prefix PR780 tag $: /* SBI ERROR REGISTER  
constant SBITA equals 53 prefix PR780 tag $: /* SBI TIMEOUT ADDRESS REGISTER  
constant SBIQC equals 54 prefix PR780 tag $: /* SBI QUADWORD CLEAR REGISTER
```

```
end_module $PR780DEF;
```



module \$PR790DEF;

{\*\*  
(\* 11/790-Specific Processor Register Definitions  
(\*-

constant	NICR	equals	25	prefix	PR790	tag	\$:	/*	INTERVAL CLOCK NEXT INTERVAL REGISTER
constant	ICR	equals	26	prefix	PR790	tag	\$:	/*	INTERVAL CLOCK INTERVAL COUNT REGISTER
constant	TODR	equals	27	prefix	PR790	tag	\$:	/*	TIME OF DAY REGISTER
constant	ACCS	equals	40	prefix	PR790	tag	\$:	/*	ACCELERATOR CONTROL STATUS REGISTER
constant	ACCR	equals	41	prefix	PR790	tag	\$:	/*	ACCELERATOR RESERVED
constant	PME	equals	61	prefix	PR790	tag	\$:	/*	PERFORMANCE MONITOR ENABLE
constant	PAMACC	equals	64	prefix	PR790	tag	\$:	/*	PAMM ACCESS
constant	PAMLOC	equals	65	prefix	PR790	tag	\$:	/*	PAMM LOCATION
constant	CSWP	equals	66	prefix	PR790	tag	\$:	/*	CACHE SWEEP REGISTER
constant	MDECC	equals	67	prefix	PR790	tag	\$:	/*	MBOX DATA ECC REGISTER
constant	MENA	equals	68	prefix	PR790	tag	\$:	/*	MBOX ERROR ENABLE REGISTER
constant	MDCTL	equals	69	prefix	PR790	tag	\$:	/*	MBOX DATA CONTROL REGISTER
constant	MCCTL	equals	70	prefix	PR790	tag	\$:	/*	MBOX MCC CONTROL REGISTER
constant	MERG	equals	71	prefix	PR790	tag	\$:	/*	MBOX ERROR GENERATOR REGISTER
constant	CRBT	equals	72	prefix	PR790	tag	\$:	/*	CONSOLE REBOOT
constant	DFI	equals	73	prefix	PR790	tag	\$:	/*	DIAGNOSTIC FAULT INSERTION
constant	EHSR	equals	74	prefix	PR790	tag	\$:	/*	ERROR HANDLING STATUS REGISTER
constant	ACCS790	equals	75	prefix	PR790	tag	\$:	/*	ACCELERATOR STATUS REGISTER
constant	STXCS	equals	76	prefix	PR790	tag	\$:	/*	CONSOLE STORAGE CONTROL REG
constant	STXDB	equals	77	prefix	PR790	tag	\$:	/*	CONSOLE STORAGE DATA REGISTER
constant	LSPA	equals	78	prefix	PR790	tag	\$:	/*	SCRATCHPAD ADDRESS
constant	RSPD	equals	79	prefix	PR790	tag	\$:	/*	SCRATCHPAD DATA

end\_module \$PR790DEF;

```
module $PRUV1DEF;
```

```
{**  
{* Micro-VAX I Processor-specific Register Definitions  
{*-
```

```
constant CADR equals 37 prefix PRUV1 tag $; /* CACHE DISABLE REGISTER  
constant MCSR equals 38 prefix PRUV1 tag $; /* MACHINE CHECK ERROR SUMMARY REG  
constant IORESET equals 55 prefix PRUV1 tag $; /* INITIALIZE BUS REGISTER
```

```
end_module $PRUV1DEF;
```



module \$PR8NDEF;

{\*\*  
{\* 11/8NN-Specific Processor Register Definitions  
{\*-

constant	NICR	equals	25	prefix	PR8NN	tag	\$:	/*	Next Interval Count Register
constant	ICR	equals	26	prefix	PR8NN	tag	\$:	/*	Interval Counter Register
constant	TODR	equals	27	prefix	PR8NN	tag	\$:	/*	Time of Year
constant	PME	equals	61	prefix	PR8NN	tag	\$:	/*	Performance Monitor Enable
constant	MCSTS	equals	38	prefix	PR8NN	tag	\$:	/*	Machine Check Status Register
constant	NMION	equals	128	prefix	PR8NN	tag	\$:	/*	NMI Interrupt Enable
constant	INOP	equals	129	prefix	PR8NN	tag	\$:	/*	Interrupt Other Processor
constant	NMIFSR	equals	130	prefix	PR8NN	tag	\$:	/*	NMI Fault/Status Register
constant	NMISILO	equals	131	prefix	PR8NN	tag	\$:	/*	NMI Bus Silo
constant	NMIEAR	equals	132	prefix	PR8NN	tag	\$:	/*	NMI Error Address Register
constant	CCR	equals	133	prefix	PR8NN	tag	\$:	/*	Cache Control Register
constant	REVR1	equals	134	prefix	PR8NN	tag	\$:	/*	Revision Register #1
constant	REVR2	equals	135	prefix	PR8NN	tag	\$:	/*	Revision Register #2

end\_module \$PR8NDEF;



module \$PR8SSDEF;

```
{**
{* 11/8SS-Specific Processor Register Definitions
{*-
```

```
constant IPIR equals 22 prefix PR8SS tag $: /* Interprocessor Interrupt Reg.
constant NICR equals 25 prefix PR8SS tag $: /* Interval Clock Next Interval Register
constant ICR equals 26 prefix PR8SS tag $: /* Interval Clock Interval Count Register
constant TODR equals 27 prefix PR8SS tag $: /* Time Of Day Register
constant TBDR equals 36 prefix PR8SS tag $: /* Translation Buffer Disable Register
constant CADR equals 37 prefix PR8SS tag $: /* Cache Disable Register
constant MCESR equals 38 prefix PR8SS tag $: /* Machine Check Error Summary Register
constant ACCS equals 40 prefix PR8SS tag $: /* Floating Point Accelerator Register
constant WCSA equals 44 prefix PR8SS tag $: /* WCS Address Register
constant WCSA equals 45 prefix PR8SS tag $: /* WCS Data Register
constant WCSC equals 46 prefix PR8SS tag $: /* WCS Cam Register
constant PME equals 61 prefix PR8SS tag $: /* Performance Monitor Enable

constant RXCS1 equals 80 prefix PR8SS tag $: /* Serial Line 1 Receive CSR
constant RXDB1 equals 81 prefix PR8SS tag $: /* Serial Line 1 Receive Data Buffer
constant TXCS1 equals 82 prefix PR8SS tag $: /* Serial Line 1 Transmit CSR
constant TXDB1 equals 83 prefix PR8SS tag $: /* Serial Line 1 Transmit Data Buffer
constant RXCS2 equals 84 prefix PR8SS tag $: /* Serial Line 2 Receive CSR
constant RXDB2 equals 85 prefix PR8SS tag $: /* Serial Line 2 Receive Data Buffer
constant TXCS2 equals 86 prefix PR8SS tag $: /* Serial Line 2 Transmit CSR
constant TXDB2 equals 87 prefix PR8SS tag $: /* Serial Line 2 Transmit Data Buffer
constant RXCS3 equals 88 prefix PR8SS tag $: /* Serial Line 3 Receive CSR
constant RXDB3 equals 89 prefix PR8SS tag $: /* Serial Line 3 Receive Data Buffer
constant TXCS3 equals 90 prefix PR8SS tag $: /* Serial Line 3 Transmit CSR
constant TXDB3 equals 91 prefix PR8SS tag $: /* Serial Line 3 Transmit Data Buffer

constant RXCD equals 92 prefix PR8SS tag $: /* Receive Console Data Register
constant CACHEX equals 93 prefix PR8SS tag $: /* Cache Invalidate Register
constant BINID equals 94 prefix PR8SS tag $: /* BI Node ID Register
constant BIINIT equals 95 prefix PR8SS tag $: /* BI Init Nodes Register
```

```
aggregate PR8SSDEF union prefix PR8SS$:
  PR8SSSID_BITS structure fill: /* Read only SID register
    SID_OCREV bitfield length 8; /* Ucode Revision Level
    SID_SECP bitfield mask; /* Secondary Patch Bit
    SID_PATREV bitfield length 10; /* Patch Rev Level
    SID_CPUREV bitfield length 5; /* CPU Rev level
    SID_TYPE bitfield length 8; /* CPU Type Code
  end PR8SSSID_BITS;

  PR8SSRXCS_BITS structure fill: /* Console RCV CSR
    FILL_T bitfield length 6 fill prefix PR8SS tag $$; /*
    RXCS_IE bitfield mask; /* Interrupt Enable
    RXCS_DONE bitfield mask; /* 1=> Char. received
  end PR8SSRXCS_BITS;

  PR8SSRXDB_BITS structure fill: /* Console RCV Data Register
    RXDB_DATA bitfield length 8; /* Received Data
    FILL_2 bitfield length 7 fill prefix PR8SS tag $$; /*
```

```

    RXDB_ERR    bitfield mask;          /* Error
end PR8SSRXDB_BITS;

PR8SSTXCS_BITS structure fill;          /* Console Transmit CSR
    FILL_3 bitfield length 6 fill prefix PR8SS tag $$; /*
    TXCS_IE    bitfield mask;          /* Interrupt Enable
    TXCS_RDY    bitfield mask;          /* Ready
    TXCS_BRE    bitfield mask;          /* (WO) Baud Rate Enable
    FILL_4 bitfield length 1 fill prefix PR8SS tag $$; /*
    TXCS_BAUD    bitfield length 3;      /* Baud Rate
                                         /* Values to set baud rates
    constant BAUD300 equals 0 prefix PR8SS tag $; /* Baud Rate of 300
    constant BAUD600 equals 1 prefix PR8SS tag $; /* Baud Rate of 600
    constant BAUD1200 equals 2 prefix PR8SS tag $; /* Baud Rate of 1200
    constant BAUD2400 equals 3 prefix PR8SS tag $; /* Baud Rate of 2400
    constant BAUD4800 equals 4 prefix PR8SS tag $; /* Baud Rate of 4800
    constant BAUD9600 equals 5 prefix PR8SS tag $; /* Baud Rate of 9600
    constant BAUD19200 equals 6 prefix PR8SS tag $; /* Baud Rate of 19200
    constant BAUD38400 equals 7 prefix PR8SS tag $; /* Baud Rate of 38400

end PR8SSTXCS_BITS;

PR8SSTXDB_BITS structure fill;          /* Console Transmit Data Register
    TXDB_DATA    bitfield length 8;      /* Data to Transmit
    TXDB_ID      bitfield length 4;      /* ID - Destination of
                                         /* transmitted data -
                                         /* 0=>UART0, F=>Console
                                         /* command
                                         /* Possible Console Commands
    constant BOOTCPU equals 2 prefix PR8SS tag $; /* Boot CPU Command
    constant CLRWARM equals 3 prefix PR8SS tag $; /* Clear Warm-start Flag
    constant CLRCOLD equals 4 prefix PR8SS tag $; /* Clear Cold-start Flag
end PR8SSTXDB_BITS;

PR8SSCADR_BITS structure fill;          /* Cache Disable Register
    CADR_D    bitfield mask;          /* Disable Cache
    CADR_H    bitfield mask;          /* Force 100% Cache Hits
end PR8SSCADR_BITS;

PR8SSWCSA_BITS structure fill;          /* WCS (Patch) Address Reg
    WCSA_DATA    bitfield length 8;      /* High Order Data Bits
    FILL_4 bitfield length 8 fill prefix PR8SS tag $$; /*
    WCSA_RAMADR    bitfield length 16;    /* Ram Address
end PR8SSWCSA_BITS;

PR8SSWCSC_BITS structure fill;          /* WCS (Patch) CAM Reg
    FILL_5 bitfield length 8 fill prefix PR8SS tag $$; /*
    WCSC_CAMADR    bitfield length 8;      /* Cam Address
    WCSC_ROMADR    bitfield length 16;    /* Rom Address
end PR8SSWCSC_BITS;

PR8SSRXCD_BITS structure fill;          /* Receive Console Data Register
    RXCD_DATA    bitfield length 8;      /* Received Data
    RXCD_NODEID    bitfield length 4;      /* Sender's Node ID
    FILL_6 bitfield length 3 fill prefix PR8SS tag $$; /*
    RXCD_BSY    bitfield mask;          /* Set=>Data has been received

```

```
end PR8SSRXCD_BITS;

PR8SSCACHEX BITS structure fill;
  FILL 7 Bitfield length 9 fill prefix PR8SS tag $$; /* Cache Invalidate Register
  CACHEX PFN bitfield length 21; /* Physical Page Number
end PR8SSCACHEX_BITS;

PR8SSBINID BITS structure fill; /* BI Node ID Register
  BINID NID bitfield length 4; /* BI Node ID this node
end PR8SSBINID_BITS;

end PR8SSDEF;
end_module $PR8SSDEF;
```



```
module $PSLDEF;
```

```
/*+
/* PROCESSOR STATUS LONGWORD MASK AND FIELD DEFINITIONS
/*-
```

```
aggregate PSLDEF union prefix PSL$;
```

```
PSLDEF BITS structure fill;
```

```

C bitfield mask;          /* Carry
V bitfield mask;          /* overflow
Z bitfield mask;          /* Zero
N bitfield mask;          /* Negative
TBIT bitfield mask;       /* TBIT ENABLE
IV bitfield mask;         /* INTEGER OVERFLOW
FU bitfield mask;         /* FLOATING UNDEFINED
DV bitfield mask;         /* DIVIDE BY ZERO
FILL 1 bitfield length 8 fill prefix PSLDEF tag $$; /*
IPL bitfield mask length 5; /* INTERRUPT PRIORITY LEVEL
FILL 2 bitfield fill prefix PSLDEF tag $$; /*
PRVMOD bitfield mask length 2; /* PREVIOUS PROCESSOR MODE
CURMOD bitfield mask length 2; /* CURRENT PROCESSOR MODE
IS bitfield mask;         /* INTERRUPT STACK BIT
FPD bitfield mask;        /* FIRST PART DONE
FILL 3 bitfield length 2 fill prefix PSLDEF tag $$; /* MUST BE ZERO
TP bitfield mask;         /* TRACE TRAP PENDING
CM bitfield mask;         /* COMPATIBILITY MODE BIT AND MASK

```

```
end PSLDEF_BITS;
```

```
/*
/* MODE SYMBOL DEFINITIONS
/*-
```

```

constant(
    KERNEL          /* KERNEL MODE
    , EXEC          /* EXEC MODE
    , SUPER         /* SUPERVISOR MODE
    , USER          /* USER MODE
    ) equals 0 increment 1 prefix PSL tag $C;

```

```

constant SAFBITS equals
( (( - (PSL$M_TP!
  PSL$M_CM!
  PSL$M_FPD)@(-16)) ) - 1 )
prefix PSL tag $M;

```

```
end PSLDEF;
```

```
end_module $PSLDEF;
```



